J 6

```
MMM         MMM   TTTTTTTTTTTTTTT     AAAAAAAAA        AAAAAAAAA        CCCCCCCCCCC   PPPPPPPPPPP
MMM         MMM   TTTTTTTTTTTTTTT     AAAAAAAAA        AAAAAAAAA        CCCCCCCCCCC   PPPPPPPPPPP
MMM         MMM   TTTTTTTTTTTTTTT     AAAAAAAAA        AAAAAAAAA        CCCCCCCCCCC   PPPPPPPPPPP
MMMMMM   MMMMMM         TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMMMMM   MMMMMM         TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMMMMM   MMMMMM         TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM   MMM   MMM         TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM   MMM   MMM         TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM   MMM   MMM         TTT        AAA         AAA  AAA         AAA  CCC              PPP         PPP
MMM         MMM         TTT        AAA         AAA  AAA         AAA  CCC              PPPPPPPPPPPP
MMM         MMM         TTT        AAA         AAA  AAA         AAA  CCC              PPPPPPPPPPPP
MMM         MMM         TTT        AAA         AAA  AAA         AAA  CCC              PPPPPPPPPPPP
MMM         MMM         TTT        AAAAAAAAAAAAAAA  AAAAAAAAAAAAAAA  CCC              PPP
MMM         MMM         TTT        AAAAAAAAAAAAAAA  AAAAAAAAAAAAAAA  CCC              PPP
MMM         MMM         TTT        AAAAAAAAAAAAAAA  AAAAAAAAAAAAAAA  CCC              PPP
MMM         MMM         TTT        AAA         AAA  AAA         AAA  CCC              PPP
MMM         MMM         TTT        AAA         AAA  AAA         AAA  CCC              PPP
MMM         MMM         TTT        AAA         AAA  AAA         AAA  CCC              PPP
MMM         MMM         TTT        AAA         AAA  AAA         AAA     CCCCCCCCCCCC  PPP
MMM         MMM         TTT        AAA         AAA  AAA         AAA     CCCCCCCCCCCC  PPP
MMM         MMM         TTT        AAA         AAA  AAA         AAA     CCCCCCCCCCCC  PPP
```

```
**FILE**ID**CHKACC                                          I 14
```

```
  CCCCCCCC  HH      HH  KK      KK    AAAAAA      CCCCCCCC    CCCCCCCC
  CCCCCCCC  HH      HH  KK      KK    AAAAAA      CCCCCCCC    CCCCCCCC
CC          HH      HH  KK      KK  AA      AA  CC          CC
CC          HH      HH  KK      KK  AA      AA  CC          CC
CC          HH      HH  KK    KK    AA      AA  CC          CC
CC          HH      HH  KK    KK    AA      AA  CC          CC
CC          HHHHHHHHHH  KKKKKK      AA      AA  CC          CC
CC          HHHHHHHHHH  KKKKKK      AA      AA  CC          CC
CC          HH      HH  KK    KK    AAAAAAAAAA  CC          CC
CC          HH      HH  KK    KK    AAAAAAAAAA  CC          CC
CC          HH      HH  KK      KK  AA      AA  CC          CC              ....
CC          HH      HH  KK      KK  AA      AA  CC          CC              ....
  CCCCCCCC  HH      HH  KK      KK  AA      AA    CCCCCCCC    CCCCCCCC      ....
  CCCCCCCC  HH      HH  KK      KK  AA      AA    CCCCCCCC    CCCCCCCC      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE CHKACC (LANGUAGE (BLISS32) ,
0                IDENT = 'V04-000'
0                ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  MTAACP
1 !
1 ! ABSTRACT:
1 !       This routine checks that the access requested is allowed on the
1 !       volume set.
1 !
1 ! ENVIRONMENT:
1 !
1 !       Starlet operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1
1 !
1 !
1 !
1 ! AUTHOR:  D. H. Gillespie,      CREATION DATE: 17-MAY-77  09:30
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-006 LMP0246         L. Mark Pilant,         2-May-1984  10:10
1 !               Correct a bug introduced by LMP0221.  The UCB and PCB addresses
1 !               were swapped in the EXE$CHKxxxACCES routine calls.
1 !
1 !       V03-005 MMD0286         Meg Dumont,     10-Apr-1984  14:14
1 !               Fix to the $MTACCESS returns where ACCESS might get
1 !               over written with a success code before all the
1 !               error conditions were checked. Fix to set the VCB
```

```
1 !                 FIL_ACCESS bit in KERNEL mode.
1 !
1 !         V03-004 LMP0221         L. Mark Pilant,         28-Mar-1984  10:21
1 !                 Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
1 !                 ORB$W_PROT.
1 !
1 !         V03-003 MMD0274         Meg Dumont,     23-Mar-1984  9:48
1 !                 Change the processing of the accessibility character fields
1 !                 in the HDR1 label to call the installation
1 !                 specific accessibility routine. The return from this
1 !                 routine determines the users access to the file. This
1 !                 module has also been changed to support the bit
1 !                 VCB$V_FIL_ACCESS which is set to determine whether
1 !                 VMS protection is valid for the file.
1 !
1 !         V03-002 MMD0239         Meg Dumont,     21-Feb-1984  10:11
1 !                 Change calls to EXE$CHKxxxACCES to kernel mode calls.
1 !
1 !         V03-001 MMD0150         Meg Dumont,     26-Apr-1983  8:51
1 !                 Change reference to 80 to the symbol ANSI_LBLSZ. Change
1 !                 reference to 240 to the symbol SCRATCH_OFFSET.
1 !
1 !         V02-007 DMW00032        David Micahel Walp      18-Aug-1981
1 !                 Looked at MVL Override Bit when override option is used
1 !
1 !         V02-006 REFORMAT        Maria del C. Nasr       30-Jun-1980
1 !
1 !         A0005   MCN0003         Maria del C. Nasr       15-Oct-1979   9:29
1 !                 Add HDR3 processing
1 !
1 !         A0004   MCN0001         Maria del C. Nasr       13-Sep-79   11:05
1 !                 Corrected bug in "create if" function
1 !
1 !**
1
1 LIBRARY 'SYS$LIBRARY:LIB.L32';
1
1 REQUIRE 'SRC$:MTADEF.B32';
1
1 FORWARD ROUTINE
1     CHECK_ACCESS          : COMMON_CALL NOVALUE,          ! check access
1     CHECK_FILE_ACC        : COMMON_CALL NOVALUE,  ! check access to file
1     CHECK_WRITE_ACCESS    : COMMON_CALL,          ! check users' write access
1     CHECK_READ_ACCESS     : COMMON_CALL,          ! check users' write access
1     SET_FILE_ACCESS       : COMMON_CALL NOVALUE,  ! Set VCB file access
1     RECALC_ST_REC         : COMMON_CALL NOVALUE;  ! recalculate start record
1
1     LINKAGE
1         CHECK_PROT        = JSB (REGISTER = 4, REGISTER = 5) :
1                                   NOPRESERVE (1, 2, 3);
1
1     EXTERNAL ROUTINE
1         EXE$CHKWRTACCES : ADDRESSING_MODE (ABSOLUTE) CHECK_PROT,
1         EXE$CHKRDACCES  : ADDRESSING_MODE (ABSOLUTE) CHECK_PROT,
1         GET_RECORD;                               ! get current record tape is reading
1
1     EXTERNAL
```

```
;  115        0498  1         CURRENT_UCB      : REF BBLOCK,           ! address of current ucb
;  116        0499  1         IO_PACKET        : REF BBLOCK,           ! address of current io request packet
;  117        0500  1         USER_STATUS      : WORD;          ! address of status to return to user
;  118        0501  1
```

```
1 GLOBAL ROUTINE CHECK_ACCESS (ACCESS_TYPE) : COMMON_CALL NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine checks that the access requested is allowed on the
1 !       volume set.
1 !
1 ! CALLING SEQUENCE:
1 !       CHECK_ACCESS(ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1 - access requested (0=read,1=write)
1 !
1 ! IMPLICIT INPUTS:
1 !       IO_PACKET       - address of current i/o packet
1 !       CURRENT_UCB     - address of current ucb
1 !
1 ! OUTPUT PARAMETERS:
1 !       None
1 !
1 ! IMPLICIT OUTPUTS:
1 !       None
1 !
1 ! ROUTINE VALUE:
1 !       None
1 !
1 ! SIDE EFFECTS:
1 !       None
1 ! USER ERROR:
1 !       SS$_WRITLCK - software write lock
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     LOCAL
2         STATUS;                                 ! io status
2
2     ! If file is software write locked and the user requests write privileges,
2     ! deny privilege
2
2     IF .ACCESS_TYPE
2         AND
2         .BBLOCK[CURRENT_UCB[UCB$L_DEVCHAR], DEV$V_SWL]
2     THEN
2         ERR_EXIT(SS$_WRITLCK);
2
2     ! If the VCB$V_FIL_ACCESS is set then the user has complete
2     ! access to this file, regardless of how the VMS protection is
2     ! set. Else check the users read adn write access to the file.
2
2     IF NOT .CURRENT_VCB[VCB$V_FIL_ACCESS]
2         THEN
```

```
;  177          0559  3              BEGIN
;  178          0560  3                  IF .ACCESS_TYPE
;  179          0561  3                  THEN
;  180          0562  4                      STATUS = KERNEL_CALL (CHECK_WRITE_ACCESS)
;  181          0563  3                  ELSE
;  182          0564  3                      STATUS = KERNEL_CALL (CHECK_READ_ACCESS);
;  183          0565  3
;  184          0566  3                  IF NOT .STATUS
;  185          0567  3                  THEN
;  186          0568  4                  BEGIN
;  187          0569  4                      USER_STATUS = .STATUS<0, 16>;
;  188          0570  4                      ERR_EXIT();
;  189          0571  3                  END;
;  190          0572  2              END;
;  191          0573  2
;  192          0574  1          END;                                            ! end of routine

                                                                 .TITLE  CHKACC
                                                                 .IDENT  \V04-000\

                                                                 .EXTRN  EXE$CHKWRTACCES
                                                                 .EXTRN  EXE$CHKRDACCES, GET_RECORD
                                                                 .EXTRN  CURRENT_UCB, IO_PACKET
                                                                 .EXTRN  USER_STATUS, SYS$CMKRNL

                                                                 .PSECT  $CODE$,NOWRT,2

                                            0000 00000          .ENTRY  CHECK_ACCESS, Save nothing           ; 0502
                   0E        04   AC     E9 00002          BLBC    ACCESS_TYPE, 1$                        ; 0547
                   50     0000G   CF     D0 00006          MOVL    CURRENT_UCB, R0                        ; 0549
       04      3B  A0                01     E1 0000B          BBC     #1, 59(R0), 1$                         ;
                           025C   8F     BF 00010          CHMU    #604                                   ; 0551
       27      2D  AB                06     E0 00014  1$:     BBS     #6, 45(CURRENT_VCB), 4$                ; 0557
                   0A        04   AC     E9 00019          BLBC    ACCESS_TYPE, 2$                        ; 0560
                                  7E     D4 0001D          CLRL    -(SP)                                  ; 0562
                                  5E     DD 0001F          PUSHL   SP                                     ;
                          0000V   CF     9F 00021          PUSHAB  CHECK_WRITE_ACCESS                     ;
                                  08     11 00025          BRB     3$                                     ;
                                  7E     D4 00027  2$:     CLRL    -(SP)                                  ; 0564
                                  5E     DD 00029          PUSHL   SP                                     ;
                          0000V   CF     9F 0002B          PUSHAB  CHECK_READ_ACCESS                      ;
            00000000G  9F                03     FB 0002F  3$:     CALLS   #3, @#SYS$CMKRNL                       ;
                       07                50     E8 00036          BLBS    STATUS, 4$                             ; 0566
                0000G  CF                50     B0 00039          MOVW    STATUS, USER_STATUS                    ; 0569
                                  00     BF 0003E          CHMU    #0                                     ; 0570
                                         04 00040  4$:     RET                                            ; 0574

; Routine Size:  65 bytes,    Routine Base:  $CODE$ + 0000


;  193          0575  1
;  195          0576  1  ROUTINE CHECK_WRITE_ACCESS : COMMON_CALL =
;  196          0577  1
;  197          0578  1  !++
;  198          0579  1  !
;  199          0580  1  ! FUNCTIONAL DESCRIPTION:
;  200          0581  1  !       This routine calls the system routine to check users write access.
;  201          0582  1  !
;  202          0583  1  ! CALLING SEQUENCE:
;  203          0584  1  !       CHECK_WRITE_ACCESS (), called in kernel mode
;  204          0585  1  !
;  205          0586  1  ! INPUT PARAMETERS:
;  206          0587  1  !       none
;  207          0588  1  !
;  208          0589  1  ! IMPLICIT INPUTS:
;  209          0590  1  !       CURRENT_UCB - address of tapes ucb
;  210          0591  1  !       IO_PACKET - address of current io request
;  211          0592  1  !
;  212          0593  1  ! OUTPUT PARAMETERS:
;  213          0594  1  !       None
;  214          0595  1  !
;  215          0596  1  ! IMPLICIT OUTPUTS:
;  216          0597  1  !       None
;  217          0598  1  !
;  218          0599  1  ! ROUTINE VALUE:
;  219          0600  1  !       STATUS from call
;  220          0601  1  !
;  221          0602  1  ! SIDE EFFECTS:
;  222          0603  1  !
;  223          0604  1  ! USER ERROR:
;  224          0605  1  !
;  225          0606  1  !--
;  226          0607  1
;  227          0608  2      BEGIN
;  228          0609  2
;  229          0610  2      EXTERNAL REGISTER
;  230          0611  2          COMMON_REG;
;  231          0612  2
;  232          0613  2      LOCAL
;  233          0614  2          PCB             : REF BBLOCK;    ! address of user process control block
;  234          0615  2
;  235          0616  2      EXTERNAL
;  236          0617  2          SCH$GL_PCBVEC   : REF VECTOR ADDRESSING_MODE (ABSOLUTE);
;  237          0618  2                                          ! system PCB vector
;  238          0619  2
;  239          0620  2          PCB = .SCH$GL_PCBVEC[.(IO_PACKET[IRP$L_PID])<0,16>];
;  240          0621  2
;  241          0622  2          RETURN EXE$CHKWRTACCES(.PCB, .CURRENT_UCB);
;  242          0623  1      END;


                                                        .EXTRN  SCH$GL_PCBVEC

                                   07FC 00000 CHECK_WRITE_ACCESS:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10          ; 0576
                     51 00000000G  9F  D0 00002         MOVL    @#SCH$GL_PCBVEC, R1                       ; 0620
                     50     0000G  CF  D0 00009         MOVL    IO_PACKET, R0                             ;
```

```
                        50          0C  C0 0000E        ADDL2   #12, R0                      ;
                        50          60  3C 00011        MOVZWL  (R0), R0                     ;
                        54        6140  D0 00014        MOVL    (R1)[R0], PCB                ;
                        55      0000G   CF  D0 00018     MOVL    CURRENT_UCB, R5              ; 0622
                           00000000G   9F  16 0001D     JSB     @#EXE$CHKWRTACCES            ;
                                           04 00023     RET                                  ; 0623

; Routine Size:  36 bytes,    Routine Base:  $CODE$ + 0041
;  244          0624  1 ROUTINE CHECK_READ_ACCESS : COMMON_CALL =
;  245          0625  1
;  246          0626  1 !++
;  247          0627  1 !
;  248          0628  1 ! FUNCTIONAL DESCRIPTION:
;  249          0629  1 !       This routine returns users' read access to the file.
;  250          0630  1 !
;  251          0631  1 ! CALLING SEQUENCE:
;  252          0632  1 !       CHECK_READ_ACCESS(), called in kernel mode
;  253          0633  1 !
;  254          0634  1 ! INPUT PARAMETERS:
;  255          0635  1 !       none
;  256          0636  1 !
;  257          0637  1 ! IMPLICIT INPUTS:
;  258          0638  1 !       CURRENT_UCB - address of tapes ucb
;  259          0639  1 !       IO_PACKET - address of current io request
;  260          0640  1 !
;  261          0641  1 ! OUTPUT PARAMETERS:
;  262          0642  1 !       None
;  263          0643  1 !
;  264          0644  1 ! IMPLICIT OUTPUTS:
;  265          0645  1 !       None
;  266          0646  1 !
;  267          0647  1 ! ROUTINE VALUE:
;  268          0648  1 !       STATUS from call
;  269          0649  1 !
;  270          0650  1 ! SIDE EFFECTS:
;  271          0651  1 !
;  272          0652  1 ! USER ERROR:
;  273          0653  1 !
;  274          0654  1 !--
;  275          0655  1
;  276          0656  2     BEGIN
;  277          0657  2
;  278          0658  2     EXTERNAL REGISTER
;  279          0659  2         COMMON_REG;
;  280          0660  2
;  281          0661  2     LOCAL
;  282          0662  2         PCB             : REF BBLOCK;   ! address of user process control block
;  283          0663  2
;  284          0664  2     EXTERNAL
;  285          0665  2         SCH$GL_PCBVEC   : REF VECTOR ADDRESSING_MODE (ABSOLUTE);
;  286          0666  2                                 ! system PCB vector
;  287          0667  2
;  288          0668  2         PCB = .SCH$GL_PCBVEC[.(IO_PACKET[IRP$L_PID])<0,16>];
;  289          0669  2
;  290          0670  2         RETURN EXE$CHKRDACCES(.PCB, .CURRENT_UCB);
;  291          0671  1     END;



                                   07FC 00000 CHECK_READ_ACCESS:
                                                     .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10                 ; 0624
                     51 00000000G  9F  D0 00002      MOVL    @#SCH$GL_PCBVEC, R1                               ; 0668
                     50      0000G  CF  D0 00009      MOVL    IO_PACKET, R0                                     ;
```

```
CHKACC                                               E 15
V04-000                                              16-Sep-1984 02:09:19    VAX-11 Bliss-32 V4.0-742              Page   9
                                                     14-Sep-1984 12:46:34    [MTAACP.SRC]CHKACC.B32;1                    (4)

                              50          0C  C0 0000E         ADDL2    #12, R0                                     ;
                              50          60  3C 00011         MOVZWL   (R0), R0                                    ;
                              54        6140  D0 00014         MOVL     (R1)[R0], PCB                               ;
                              55      0000G   CF  D0 00018     MOVL     CURRENT_UCB, R5                             ; 0670
                                  00000000G   9F  16 0001D     JSB      @#EXE$CHKRDACCES                            ;
                                                  04 00023     RET                                                  ; 0671

; Routine Size:  36 bytes,    Routine Base:  $CODE$ + 0065


;  292          0672  1
;  293          0673  1
```

```
1 GLOBAL ROUTINE CHECK_FILE_ACC (ACCESS_CALL) : COMMON_CALL NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine checks access to the file.  If accessibility code is
1 !       not blank and not overridden then access is denied.  If writing to the file the file
1 !       must be expired.
1 !
1 ! CALLING SEQUENCE:
1 !       CHECK_FILE_ACC(ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       0 - If being called from MTA_CREATE
1 !       1 - If called from MTA_ACCESS
1 !
1 ! IMPLICIT INPUTS:
1 !       LOCAL_FIB - copy of user's file information block
1 !       CURRENT_VCB - address of current control block
1 !
1 ! OUTPUT PARAMETERS:
1 !       None
1 !
1 ! IMPLICIT OUTPUTS:
1 !       None
1 !
1 ! ROUTINE VALUE:
1 !       None
1 !
1 ! SIDE EFFECTS:
1 !       if append, tape is positioned to end of data
1 !
1 ! USER ERROR:
1 !       SS$_FILACCERR - file access byte non-blank
1 !
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         COMMON_REG;
2
2     EXTERNAL ROUTINE
2         EXPIRED         : COMMON_CALL,          ! check that file has expired
2         LIB$CVT_DTB     : ADDRESSING_MODE (ABSOLUTE),
2                                                 ! convert decimal to binary
2         SPACE_EOF       : COMMON_CALL,          ! space to trailers
2         SPACE_TM        : COMMON_CALL,          ! space given number of
2                                                 !   tape marks
2         READ_BLOCK      : COMMON_CALL;          ! read on mag tape data block
2
2     LOCAL
2         ACCESS,                                 ! users' access to the file
2         BLOCK_COUNT,                            ! block count of file to
2                                                 !  appended to
2         CURRENT_RECORD,                         ! record  tape drive is reading
2         FIB             : REF BBLOCK,           ! address of local fib
```

```
  2          LABELADDR        : REF BBLOCK,            ! address of label
  2          STATUS,
  2          TM,                                       ! number of tm's
  2          ORB              : REF BBLOCK,            ! ORB address
  2          MVL              : REF BBLOCK,
  2          MVL_ENTRY        : REF BBLOCKVECTOR [,MVL$K_LENGTH];
  2                                                    ! pointer to the MVL_ENTRY
  2
  2      EXTERNAL
  2          HDR1             : REF BBLOCK,            ! address of HDR1(EOF1) label
  2          LOCAL_FIB        : BBLOCK;                ! copy of user's fib
  2
  2      ! setup pointer to fib
  2
  2      FIB = LOCAL_FIB;
  2
  2      ! get a handle on the MVL entry
  2
  2      MVL = .CURRENT_VCB[VCB$L_MVL];
  2      MVL_ENTRY = (.CURRENT_VCB [ VCB$L_MVL ]) +  MVL$K_FIXLEN;
  2
  2      ! Call the accessibility system service to check the accessibility char
  2      ! on the HDR1 label.
  2      ! First keep the record that the UCB is reading. The accessibility
  2      ! routine can not move the tape from under us! Thus we will compare
  2      ! this to the field after the call and if the tape was moved we punt
  2      ! the operation.
  2
  2     ORB = .CURRENT_UCB[UCB$L_ORB];
  2     CURRENT_RECORD = KERNEL_CALL(GET_RECORD, .CURRENT_UCB);
P 2     ACCESS = $MTACCESS(LBLNAM = .HDR1,
P 2                     UIC = .ORB[ORB$L_OWNER],
P 2                     STD_VERSION = .MVL[MVL$B_STDVER],
P 2                     ACCESS_CHAR = 0,
P 2                     ACCESS_SPEC = MTA$K_NOCHAR,
  2                     TYPE = MTA$K_INHDR1);
  2
  2     STATUS = KERNEL_CALL(GET_RECORD, .CURRENT_UCB);
  2     IF .CURRENT_RECORD NEQ .STATUS
  2          THEN ERR_EXIT(SS$_TAPEPOSLOST);
  2
  2     IF .ACCESS EQL SS$_FILACCERR
  2          THEN
  3          BEGIN
  4              IF NOT (      .CURRENT_VCB[VCB$V_OVRACC]
  4                     AND .MVL_ENTRY[ (.CURRENT_VCB[VCB$W_RVN]), MVL$V_OVERIDE ])
  3                  THEN ERR_EXIT(SS$_FILACCERR);
  3              ACCESS = SS$_NORMAL;
  2          END;
  2
  2     IF .ACCESS EQL SS$_NOVOLACC OR .ACCESS EQL SS$_NOFILACC
  2          THEN ERR_EXIT(.ACCESS);
  2
  2     IF NOT .ACCESS THEN KERNEL_CALL(SET_FILE_ACCESS);
  2
  2     ! now treat append case uniquely
```

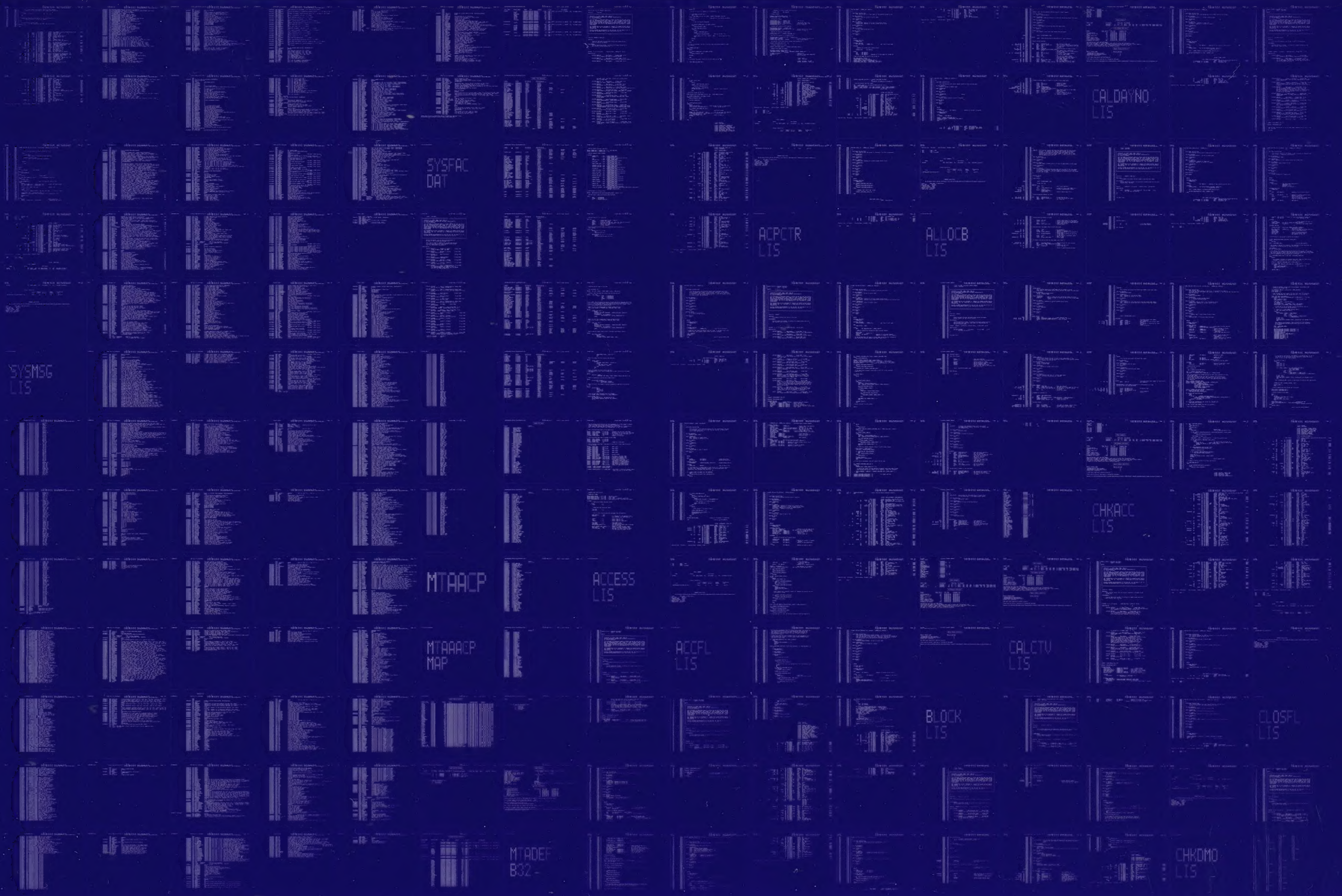
```
IF ( NOT .FIB[FIB$V_UPDATE])
     AND
     .FIB[FIB$V_WRITE]
     AND
     (.ACCESS_CALL)
     THEN
     BEGIN                                          ! append case
         SPACE_EOF();                               ! left at absolute end of file

         IF NOT LIB$CVT_DTB(EO1$S_BLOCKCNT, HDR1[EO1$T_BLOCKCNT], BLOCK_COUNT)
              THEN
                  ERR_EXIT(SS$_BLOCKCNTERR);

         ! read header of next file

         LABELADDR = .HDR1 + SCRATCH_OFFSET;  ! read into scratch area

         IF NOT READ_BLOCK(.LABELADDR, ANSI_LBLSZ)
              THEN
              BEGIN                                 ! at logical end of volume set
                  SPACE_TM(-3);        ! double tape mark is logical end of tape
                  KERNEL_CALL(RECALC_ST_REC, .BLOCK_COUNT);
                  RETURN;
              END;

        IF .LABELADDR[HD1$L_HD1LID] NEQ 'HDR1'
              THEN ERR_EXIT(SS$_TAPEPOSLOST);

        ! going to overlay file

        IF NOT EXPIRED(LABELADDR[HD1$T_EXPIREDT])
              THEN ERR_EXIT(SS$_FILNOTEXP);

        SPACE_TM(-2);
        KERNEL_CALL(RECALC_ST_REC, .BLOCK_COUNT);
        RETURN;

        END;                                        ! end of append case

 ! if about to write current file check expiration

 IF .FIB[FIB$V_WRITE]
 THEN

     IF NOT EXPIRED(HDR1[HD1$T_EXPIREDT])
     THEN
         ERR_EXIT(SS$_FILNOTEXP);

 END;                                               ! end of routine


                                                      .EXTRN  EXPIRED, LIB$CVT_DTB
                                                      .EXTRN  SPACE_EOF, SPACE_TM
                                                      .EXTRN  READ_BLOCK, HDR1
                                                      .EXTRN  LOCAL_FIB, SYS$MTACCESS
```

```
                                    01FC 00000        .ENTRY   CHECK_FILE_ACC, Save R2,R3,R4,R5,R6,R7,R8   ; 0674
                58     0000G    CF    9E 00002        MOVAB    HDR1, R8                                    ;
                57 00000000G    9F    9E 00007        MOVAB    @#SYS$CMKRNL, R7                            ;
                5E              04    C2 0000E        SUBL2    #4, SP                                      ;
                55     0000G    CF    9E 00011        MOVAB    LOCAL_FIB, FIB                              ; 0745
                52        34    AB    D0 00016        MOVL     52(CURRENT_VCB), MVL                        ; 0749
53          34  AB              24    C1 0001A        ADDL3    #36, 52(CURRENT_VCB), MVL_ENTRY             ; 0750
                50     0000G    CF    D0 0001F        MOVL     CURRENT_UCB, R0                             ; 0759
                54        1C    A0    D0 00024        MOVL     28(R0), ORB                                 ;
                                50    DD 00028        PUSHL    R0                                          ; 0760
                                01    DD 0002A        PUSHL    #1                                          ;
                                5E    DD 0002C        PUSHL    SP                                          ;
                       0000G    CF    9F 0002E        PUSHAB   GET_RECORD                                  ;
                67              04    FB 00032        CALLS    #4, SYS$CMKRNL                              ;
                56              50    D0 00035        MOVL     R0, CURRENT_RECORD                          ;
                                01    DD 00038        PUSHL    #1                                          ; 0766
                                7E    7C 0003A        CLRQ     -(SP)                                       ;
                7E        22    A2    9A 0003C        MOVZBL   34(MVL), -(SP)                              ;
                                64    DD 00040        PUSHL    (ORB)                                       ;
                                68    DD 00042        PUSHL    HDR1                                        ;
     00000000G  00              06    FB 00044        CALLS    #6, SYS$MTACCESS                            ;
                52              50    D0 0004B        MOVL     R0, ACCESS                                  ;
                       0000G    CF    DD 0004E        PUSHL    CURRENT_UCB                                 ; 0768
                                01    DD 00052        PUSHL    #1                                          ;
                                5E    DD 00054        PUSHL    SP                                          ;
                       0000G    CF    9F 00056        PUSHAB   GET_RECORD                                  ;
                67              04    FB 0005A        CALLS    #4, SYS$CMKRNL                              ;
                50              56    D1 0005D        CMPL     CURRENT_RECORD, STATUS                      ; 0769
                                04    13 00060        BEQL     1$                                          ;
                        0224    8F    BF 00062        CHMU     #548                                        ; 0770
     0000009C   8F              52    D1 00066  1$:   CMPL     ACCESS, #156                                ; 0772
                                18    12 0006D        BNEQ     4$                                          ;
0C          2C  AB              01    E1 0006F        BBC      #1, 44(CURRENT_VCB), 2$                     ; 0775
                50        0E    AB    3C 00074        MOVZWL   14(CURRENT_VCB), R0                         ; 0776
                          07  A340    7F 00078        PUSHAQ   7(MVL_ENTRY)[R0]                            ;
04              9E              02    E0 0007C        BBS      #2, @(SP)+, 3$                              ;
                        009C    8F    BF 00080  2$:   CHMU     #156                                        ; 0777
                52              01    D0 00084  3$:   MOVL     #1, ACCESS                                  ; 0778
     000022A4   8F              52    D1 00087  4$:   CMPL     ACCESS, #8868                               ; 0781
                                09    13 0008E        BEQL     5$                                          ;
     000022AC   8F              52    D1 00090        CMPL     ACCESS, #8876                               ;
                                02    12 00097        BNEQ     6$                                          ;
                                52    BF 00099  5$:   CHMU     ACCESS                                      ; 0782
                0B              52    E8 0009B  6$:   BLBS     ACCESS, 7$                                  ; 0784
                                7E    D4 0009E        CLRL     -(SP)                                       ;
                                5E    DD 000A0        PUSHL    SP                                          ;
                       0000V    CF    9F 000A2        PUSHAB   SET_FILE_ACCESS                             ;
                67              03    FB 000A6        CALLS    #3, SYS$CMKRNL                              ;
70              65              06    E0 000A9  7$:   BBS      #6, (FIB), 13$                              ; 0788
                6C        01    A5    E9 000AD        BLBC     1(FIB), 13$                                 ; 0790
                68        04    AC    E9 000B1        BLBC     ACCESS_CALL, 13$                            ; 0792
         0000G  CF              00    FB 000B5        CALLS    #0, SPACE_EOF                               ; 0795
                                5E    DD 000BA        PUSHL    SP                                          ; 0797
7E              68              36    C1 000BC        ADDL3    #54, HDR1, -(SP)                            ;
                                06    DD 000C0        PUSHL    #6                                          ;
     00000000G  9F              03    FB 000C2        CALLS    #3, @#LIB$CVT_DTB                           ;
                04              50    E8 000C9        BLBS     R0, 8$                                      ;
```

```
                               0940   8F  BF 000CC          CHMU     #2368                                         ; 0799
          53           68 00000140   8F  C1 000D0  8$:      ADDL3    #320, HDR1, LABELADDR                         ; 0803
                       7E        50   8F  9A 000D8          MOVZBL   #80, -(SP)                                    ; 0805
                                      53  DD 000DC          PUSHL    LABELADDR                                     ;
                0000G  CF             02  FB 000DE          CALLS    #2, READ_BLOCK                                ;
                       05             50  E8 000E3          BLBS     R0, 9$                                        ;
                       7E             03  CE 000E6          MNEGL    #3, -(SP)                                     ; 0808
                                      1F  11 000E9          BRB      12$                                           ;
             31524448  8F             63  D1 000EB  9$:     CMPL     (LABELADDR), #827475016                       ; 0813
                                      04  13 000F2          BEQL     10$                                           ;
                               0224   8F  BF 000F4          CHMU     #548                                          ; 0814
                                 2F   A3  9F 000F8  10$:    PUSHAB   47(LABELADDR)                                 ; 0818
                0000G  CF             01  FB 000FB          CALLS    #1, EXPIRED                                   ;
                       04             50  E8 00100          BLBS     R0, 11$                                       ;
                               00B4   8F  BF 00103          CHMU     #180                                          ; 0819
                       7E             02  CE 00107  11$:    MNEGL    #2, -(SP)                                     ; 0821
                0000G  CF             01  FB 0010A  12$:    CALLS    #1, SPACE_TM                                  ;
                                      6E  DD 0010F          PUSHL    BLOCK_COUNT                                   ; 0822
                                      01  DD 00111          PUSHL    #1                                            ;
                                      5E  DD 00113          PUSHL    SP                                            ;
                              0000V   CF  9F 00115          PUSHAB   RECALC_ST_REC                                 ;
                       67             04  FB 00119          CALLS    #4, SYS$CMKRNL                                ;
                                          04 0011C          RET                                                    ; 0794
                       10        01   A5  E9 0011D  13$:    BLBC     1(FIB), 14$                                   ; 0829
          7E           68             2F  C1 00121          ADDL3    #47, HDR1, -(SP)                              ; 0832
                0000G  CF             01  FB 00125          CALLS    #1, EXPIRED                                   ;
                       04             50  E8 0012A          BLBS     R0, 14$                                       ;
                               00B4   8F  BF 0012D          CHMU     #180                                          ; 0834
                                          04 00131  14$:    RET                                                    ; 0836

; Routine Size:  306 bytes,    Routine Base:  $CODE$ + 0089


;  458         0837  1
;  460          0838  1 ROUTINE SET_FILE_ACCESS : COMMON_CALL NOVALUE =
;  461          0839  1
;  462          0840  1 !++
;  463          0841  1 !
;  464          0842  1 ! FUNCTIONAL DESCRIPTION:
;  465          0843  1 !       This routine updates the VCB file access bit to say that the user
;  466          0844  1 !       has complete access to the file.
;  467          0845  1 !
;  468          0846  1 ! CALLING SEQUENCE:
;  469          0847  1 !       SET_FILE_ACCESS called in kernel mode
;  470          0848  1 !
;  471          0849  1 ! INPUT PARAMETERS:
;  472          0850  1 !       none
;  473          0851  1 !
;  474          0852  1 ! IMPLICIT INPUTS:
;  475          0853  1 !       CURRENT_VCB - address of current volume control block
;  476          0854  1 !
;  477          0855  1 ! OUTPUT PARAMETERS:
;  478          0856  1 !       None
;  479          0857  1 !
;  480          0858  1 ! IMPLICIT OUTPUTS:
;  481          0859  1 !       VCB FIL_ACCESS bit is set.
;  482          0860  1 !
;  483          0861  1 ! ROUTINE VALUE:
;  484          0862  1 !       None
;  485          0863  1 !
;  486          0864  1 ! SIDE EFFECTS:
;  487          0865  1 !       None
;  488          0866  1 !
;  489          0867  1 ! USER ERRORS:
;  490          0868  1 !       None
;  491          0869  1 !
;  492          0870  1 !--
;  493          0871  1
;  494          0872  2    BEGIN
;  495          0873  2
;  496          0874  2    EXTERNAL REGISTER
;  497          0875  2        COMMON_REG;
;  498          0876  2
;  499          0877  2        CURRENT_VCB[VCB$V_FIL_ACCESS] = 1;
;  500          0878  1    END;                                    ! end of routine



                                           0000 00000 SET_FILE_ACCESS:
                                                              .WORD   Save nothing                              ; 0838
                    2D   AB        40   8F   88 00002         BISB2   #64, 45(CURRENT_VCB)                      ; 0877
                                             04 00007         RET                                               ; 0878

; Routine Size:  8 bytes,    Routine Base:  $CODE$ + 01BB


;  501          0879  1
;  502          0880  1
;  504          0881   1 ROUTINE RECALC_ST_REC (BLOCK_COUNT) : COMMON_CALL NOVALUE =
;  505          0882   1
;  506          0883   1 !++
;  507          0884   1 !
;  508          0885   1 ! FUNCTIONAL DESCRIPTION:
;  509          0886   1 !       This routine updates the start record count to include those
;  510          0887   1 !       records in the file that were previously written.
;  511          0888   1 !
;  512          0889   1 ! CALLING SEQUENCE:
;  513          0890   1 !       RECALC_ST_REC(ARG1), called in kernel mode
;  514          0891   1 !
;  515          0892   1 ! INPUT PARAMETERS:
;  516          0893   1 !       ARG1 - number of blocks previously written
;  517          0894   1 !
;  518          0895   1 ! IMPLICIT INPUTS:
;  519          0896   1 !       CURRENT_VCB - address of current volume control block
;  520          0897   1 !
;  521          0898   1 ! OUTPUT PARAMETERS:
;  522          0899   1 !       None
;  523          0900   1 !
;  524          0901   1 ! IMPLICIT OUTPUTS:
;  525          0902   1 !       Start record number updated to reflect previously written records
;  526          0903   1 !
;  527          0904   1 ! ROUTINE VALUE:
;  528          0905   1 !       None
;  529          0906   1 !
;  530          0907   1 ! SIDE EFFECTS:
;  531          0908   1 !       None
;  532          0909   1 !
;  533          0910   1 ! USER ERRORS:
;  534          0911   1 !       None
;  535          0912   1 !
;  536          0913   1 !--
;  537          0914   1
;  538          0915   2     BEGIN
;  539          0916   2
;  540          0917   2     EXTERNAL REGISTER
;  541          0918   2         COMMON_REG;
;  542          0919   2
;  543          0920   2     CURRENT_VCB[VCB$L_ST_RECORD] = .CURRENT_VCB[VCB$L_ST_RECORD] -
;  544          0921   2     .BLOCK_COUNT;
;  545          0922   1     END;                                        ! end of routine



                                            0000 00000 RECALC_ST_REC:
                                                              .WORD   Save nothing                              ; 0881
                        30   AB        04   AC  C2 00002      SUBL2   BLOCK_COUNT, 48(CURRENT_VCB)              ; 0921
                                                04 00007      RET                                               ; 0922

; Routine Size:  8 bytes,    Routine Base:  $CODE$ + 01C3


;  546          0923   1 END
;  547          0924   1
;  548         0925  0 ELUDOM



;                                       PSECT SUMMARY
;
;       Name                         Bytes                          Attributes
;
;  $CODE$                                 459  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                               Library Statistics
;
;                                                  -------- Symbols --------         Pages        Processing
;       File                                       Total    Loaded   Percent         Mapped       Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1                   18619        37         0         1000           00:01.8




;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CHKACC/OBJ=OBJ$:CHKACC MSRC$:CHKACC/UPDATE=(ENH$:CHKACC)

; Size:           459 code + 0 data bytes
; Run Time:          00:13.5
; Elapsed Time:      00:51.6
; Lines/CPU Min:      4098
; Lexemes/CPU-Min:  19364
; Memory Used:  145 pages
; Compilation Complete
```

0253 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



CALDAYNO
LIS

SYSFAC
DAT

ACPCTR
LIS

ALLOCB
LIS

SYSMSG
LIS

CHKACC
LIS

MTAACP

ACCESS
LIS

MTAAACP
MAP

ACCFL
LIS

CALCTV
LIS

BLOCK
LIS

CLOSFL
LIS

MTADEF
B32

CHKDMO
LIS